



IMPORTING DATA INTO R

The screenshot shows the R Studio interface with the following components:

- Top Bar:** File, Edit, Code, View, Plots, Session, Build, Debug, Tools, Help.
- Toolbar:** Includes icons for file operations like Open, Save, Run, Source, and Addins.
- Code Editor:** A script editor titled "chicagoFood.R" containing R code to import data from a CSV file and create a Leaflet map.
- Console:** A text area showing the R code being run and its output, including the creation of a subset of data and the execution of the Leaflet code.
- Data View:** Shows two datasets: "data" (118607 obs. of 17 variables) and "data1" (50 obs. of 17 variables).
- Map View:** A Leaflet map of the Chicago area with several blue location pins placed on it, corresponding to the data points.

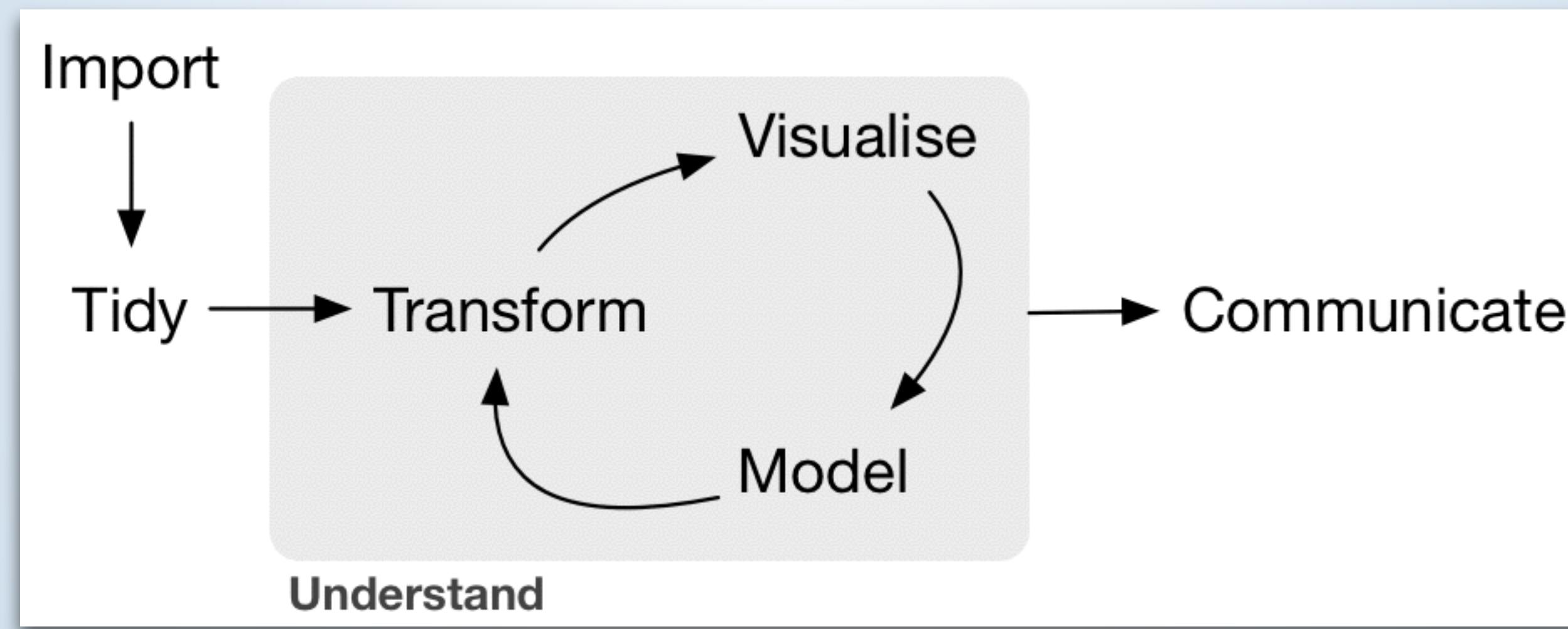
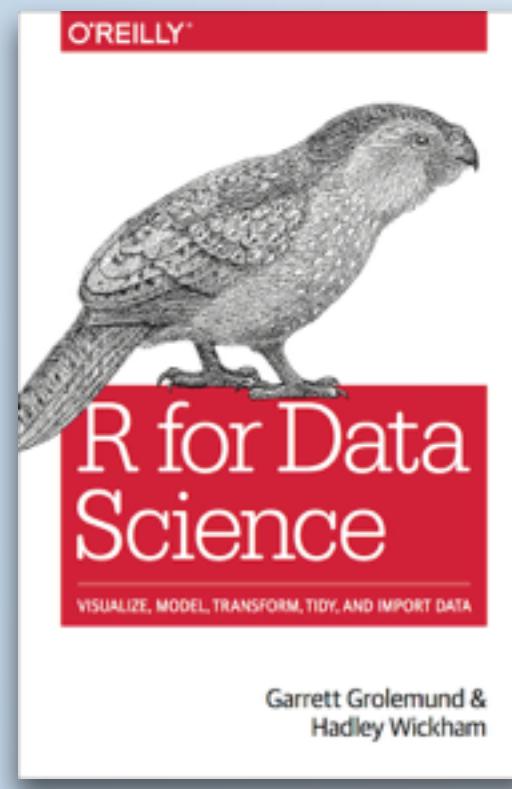
```
1 url <- "http://data.cityofchicago.org/api/views/4ijn-s7e5/rows.csv?c=&q=&method=export&format=csv"
2 data <- read.csv(url, header = TRUE) # takes a minute...
3 names(data) <- tolower(names(data))
4 data1 <- subset(data, risk %in% c("Risk 1 (High)", "Risk 2 (Medium)", "Risk 3 (Low)"))
5 data1$risk <- droplevels(data1$risk)
6
7 data1 <- data1[1:50,]
8 library(leaflet)
9 leaflet(data1) %>%
10   addTiles() %>%
11   addMarkers(lat = ~latitude, lng = ~longitude)
```

```
11:33 | (Top Level) +-----+
Console ~/Radar/ ~-----+
> data1 <- subset(data, risk %in% c("Risk 1 (High)", "Risk 2 (Medium)", "Risk 3 (Low)"))
> data1$risk <- droplevels(data1$risk)
>
> data1 <- data1[1:50,]
> library(leaflet)
> leaflet(data1) %>%
+   addTiles() %>%
+   addMarkers(lat = ~latitude, lng = ~longitude)
>
```

IMPORTING DATA INTO R

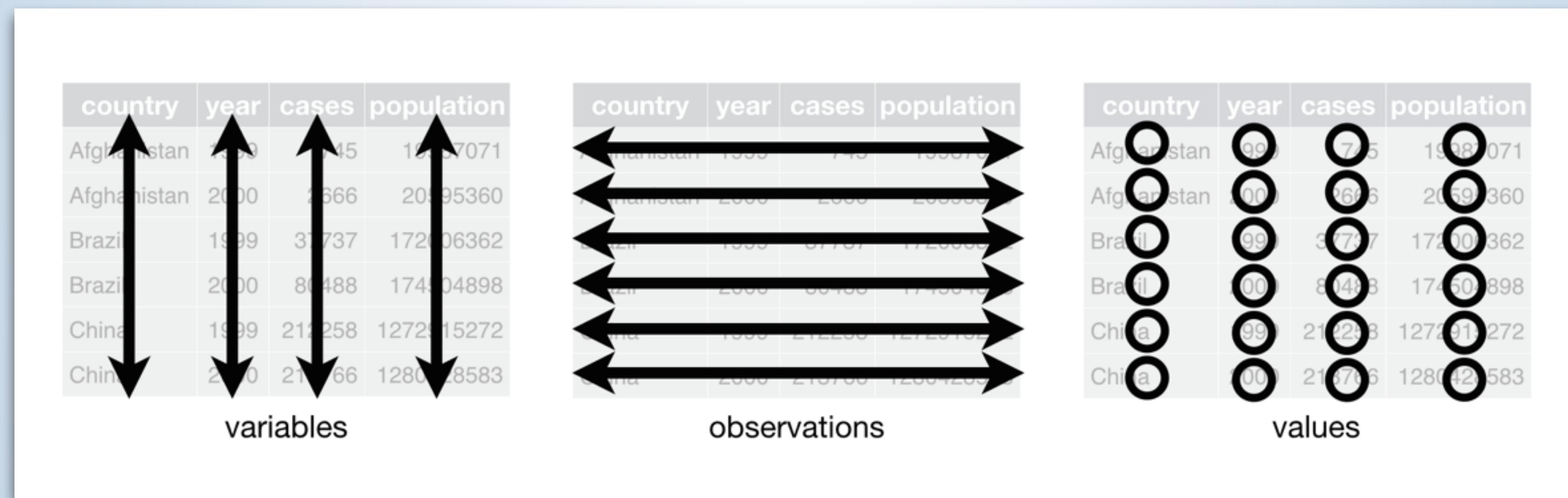
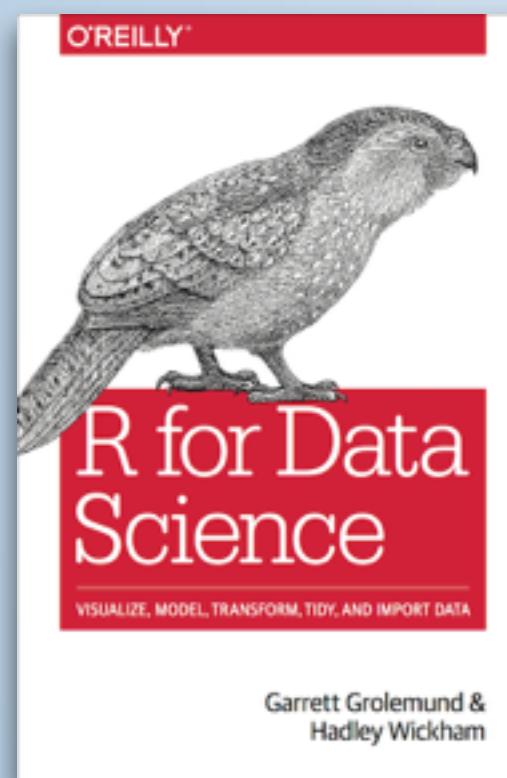
- Overview
- Importing _____ Data
 - Tabular
 - Hierarchical
 - Relational
 - Distributed
- Questions

OVERVIEW: THE TYPICAL DATA SCIENCE PROJECT



Importing Tabular Data

WHAT IS TABULAR DATA?



TEXT, EXCEL, SPSS, SAS AND STATA

```
# Import from Text
library(readr)
read_csv("inst/data/Water_Right_Applications.csv")

# Import from Excel
library(readxl)
read_excel("inst/data/Water_Right_Applications.xls")

# Import from SPSS
library(haven)
read_sav("inst/data/Child_Data.sav")

# Import from SAS
read_sas("inst/data/iris.sas7bdat")

# Import from STATA
read_dta("inst/data/Milk_Production.dta")
```



TABULAR DATA FROM RSTUDIO

The screenshot shows the RStudio interface with a search bar at the top containing "rstudio preview". The main area displays the "Import Text Data" dialog for importing a CSV file named "Water_Right_Applications_ISO2022JP.csv".

Import Options:

- Name: water_right_applicatio
- Skip: 0
- First Row as Names
- Trim Spaces
- Delimiter: Comma
- Escape: None
- Quotes: Default
- Comment: Default
- Encoding: Default
- NA: Default

Code Preview:

```
library(readr)
water_right_applications_iso2022jp <- read_csv("~/RStudio/features/import/samples/Water_Right_Applications_ISO2022JP.csv")
View(water_right_applications_iso2022jp)
```

Buttons: Import, Cancel

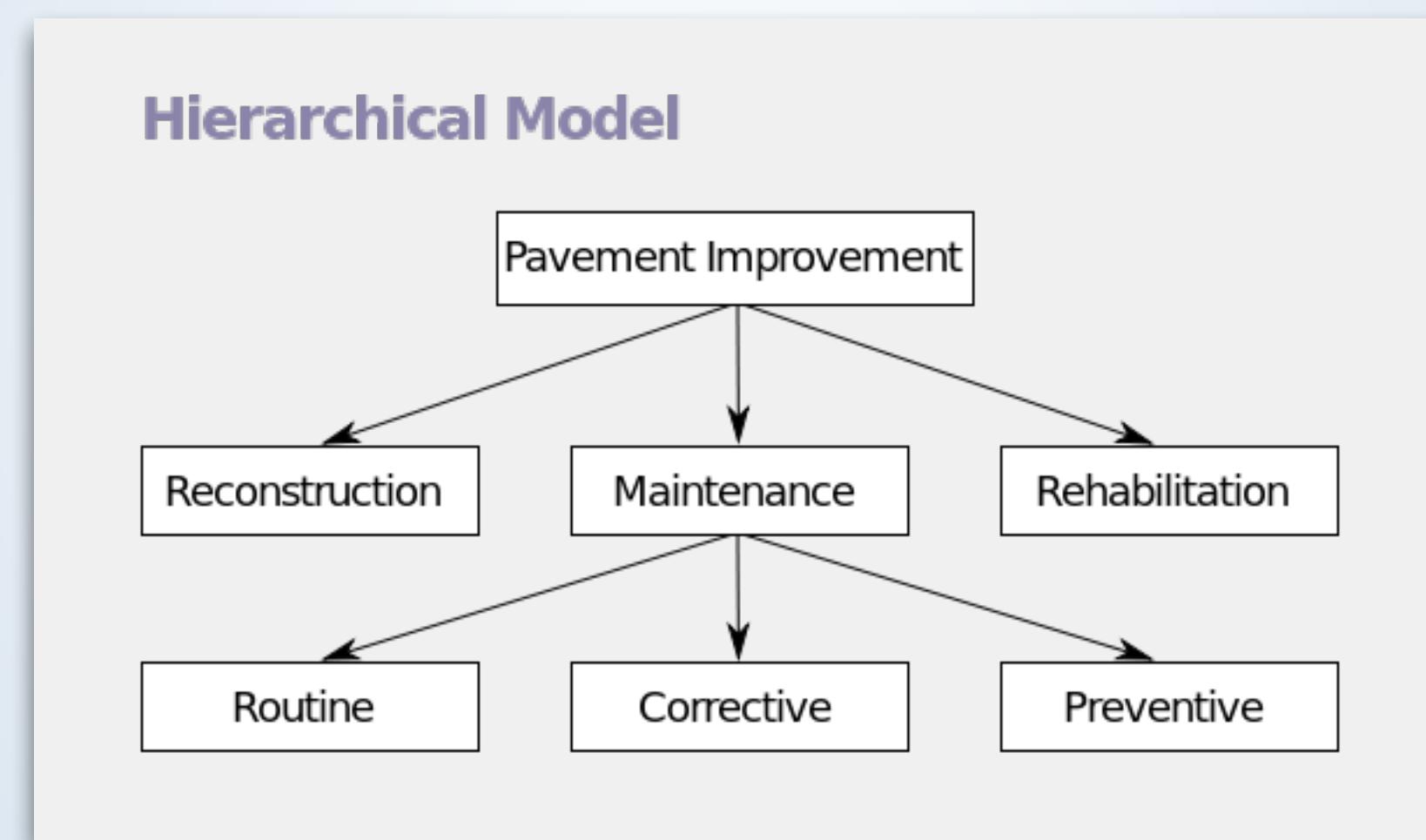
WR_DOC_ID (integer)	DOCUMENT_NUMBER (character)	DOCUMENT_TYPE (character)	PURPOSE_CODE_LIST (character)	PERSON_LAST_OR_ORGANIZATION_NAME (character)	PRIORITY_DATE (character)	YEAR
2229352	S1-*04254	NewApp	CI MU	Seattle Water Dept	07/14/1936	
2085332	R4-10948	NewApp	IR	US Bureau Reclamation - Boise	12/29/1951	
2285593	S1-13219	NewApp	MU	Everett City	12/15/1954	
2285597	G1-*12139	NewApp	DM	Sandy Point Improvement Co	08/05/1971	
2285599	G1-*12141	NewApp	DM	Sandy Point Improvement Co	08/05/1971	
2283433	G3-20099	NewApp	CI	Wapato Fruit Products	03/20/1972	
2283437	G3-20101	NewApp	CI	Wapato Fruit Products	03/20/1972	



Importing Hierarchical Data

WHAT IS HIERARCHICAL DATA?

“A data model in which the data is organized into a tree-like structure” - Wikipedia



WHAT IS XML, HTML AND JSON?

XML: Extensible Markup Language

```
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

JSON: JavaScript Object Notation

```
{"employees": [
    {"firstName": "John", "lastName": "Doe"},
    {"firstName": "Anna", "lastName": "Smith"},
    {"firstName": "Peter", "lastName": "Jones"}
]}
```

HTML: HyperText Markup Language

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>This is a Heading</h1>
    <p>This is a paragraph.</p>
  </body>
</html>
```

IMPORTING XML, HTML AND JSON

```
# Import from XML
library(xml2)
xml <- read_xml("inst/data/Water_Right_Applications.xml")
xml_children(xml_children(xml))

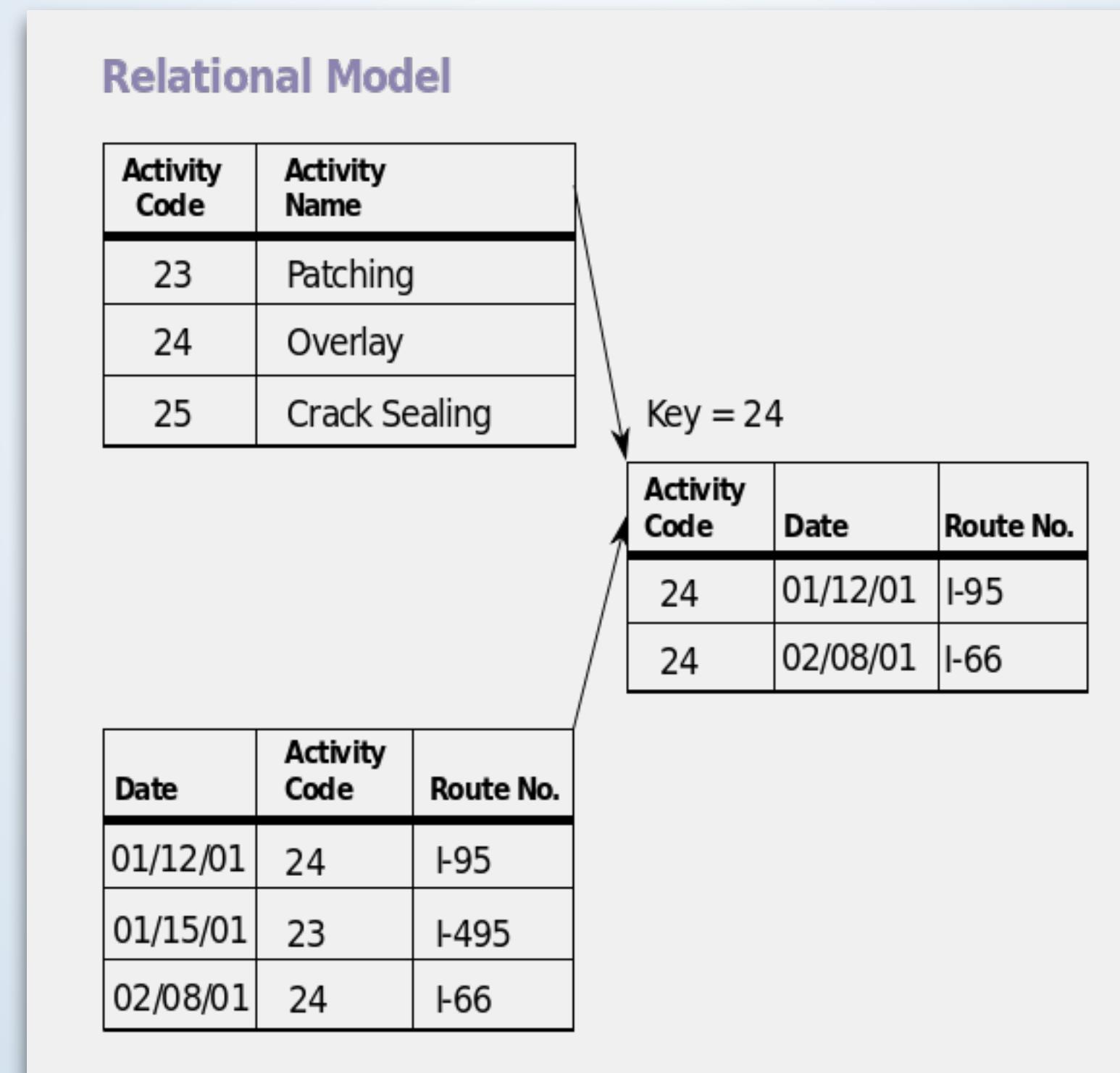
# Import from HTML
library(rvest)
html <- read_html("https://en.wikipedia.org/wiki/List_of_countries_and_dependencies_by_population")
table <- xml_find_one(html, "//table")
html_table(table)

# Import from JSON
library(jsonlite)
json <- fromJSON("inst/data/Water_Right_Applications.json")
```

Importing Relational Data

WHAT IS RELATIONAL DATA?

“Data is represented in terms of tuples, grouped into relations.” - Wikipedia



IMPORTING RELATIONAL DATA



List of Software			
<ul style="list-style-type: none">• 4th Dimension• Adabas D• Alpha Five• Apache Derby• Aster Data• Amazon Aurora• Altibase• CA Datacom• CA IDMS• Clarion• Clustrix• CSQL• CUBRID• DataEase• Database Management Library• Dataphor• dBase• Derby aka Java DB• Empress Embedded Database• EXASolution• EnterpriseDB	<ul style="list-style-type: none">• eXtremeDB• FileMaker Pro• Firebird• FrontBase• Greenplum• GroveSite• Hadoop• H2• Helix database• HSQldb• IBM DB2• IBM Lotus Approach• IBM DB2 Express-C• Infobright• Informix• Ingres• InterBase• InterSystems Caché• GT.M• LibreOffice Base• Linter	<ul style="list-style-type: none">• MariaDB• MaxDB• MemSQL• Microsoft Access• Microsoft Jet Database Engine (part of Microsoft Access)• Microsoft SQL Server• Microsoft SQL Server Express• SQL Azure (Cloud SQL Server)• Microsoft Visual FoxPro• Mimer SQL• MonetDB• mSQL• MySQL• Netezza• NexusDB• NonStop SQL• NuoDB• Openbase• OpenLink Virtuoso (Open Source Edition)	<ul style="list-style-type: none">• OpenLink Virtuoso Universal Server• OpenOffice.org Base• Oracle• Oracle Rdb for OpenVMS• Panorama• Pervasive PSQL• Polyhedra• PostgreSQL• Postgres Plus Advanced Server• Progress Software• RDM Embedded• RDM Server• R:Base• The SAS system• SAND CDBMS• SAP HANA• SAP Sybase Adaptive Server Enterprise• SAP Sybase IQ Edition <ul style="list-style-type: none">• SQL Anywhere (formerly known as Sybase Adaptive Server Anywhere and Watcom SQL)• ScimoreDB• SmallSQL• solidDB• SQLBase• SQLite• Sybase Advantage Database Server• Teradata• TimesTen• Trafodion• txtSQL• Unisys RDMS 2200• UnQLite• UniData• UniVerse• Vectorwise• Vertica• VMDS

Icons Credit: Hans Gerhard Meier and Wilson Joseph

IMPORTING RELATIONAL DATA

```
# Connect to Postgres
library(DBI)
db <- dbConnect(RPostgres::Postgres(), user, pass, ...)

# Connect to MySQL
db <- dbConnect(RMySQL::MySQL(), user, pass, ...)

# Connect to SQLite
db <- dbConnect(RSQLite::SQLite(), dbname = "inst/data/database.sqlite")

# Import data from SQLite
dbListTables(db)
View(dbGetQuery(db, "SELECT * FROM packages"))

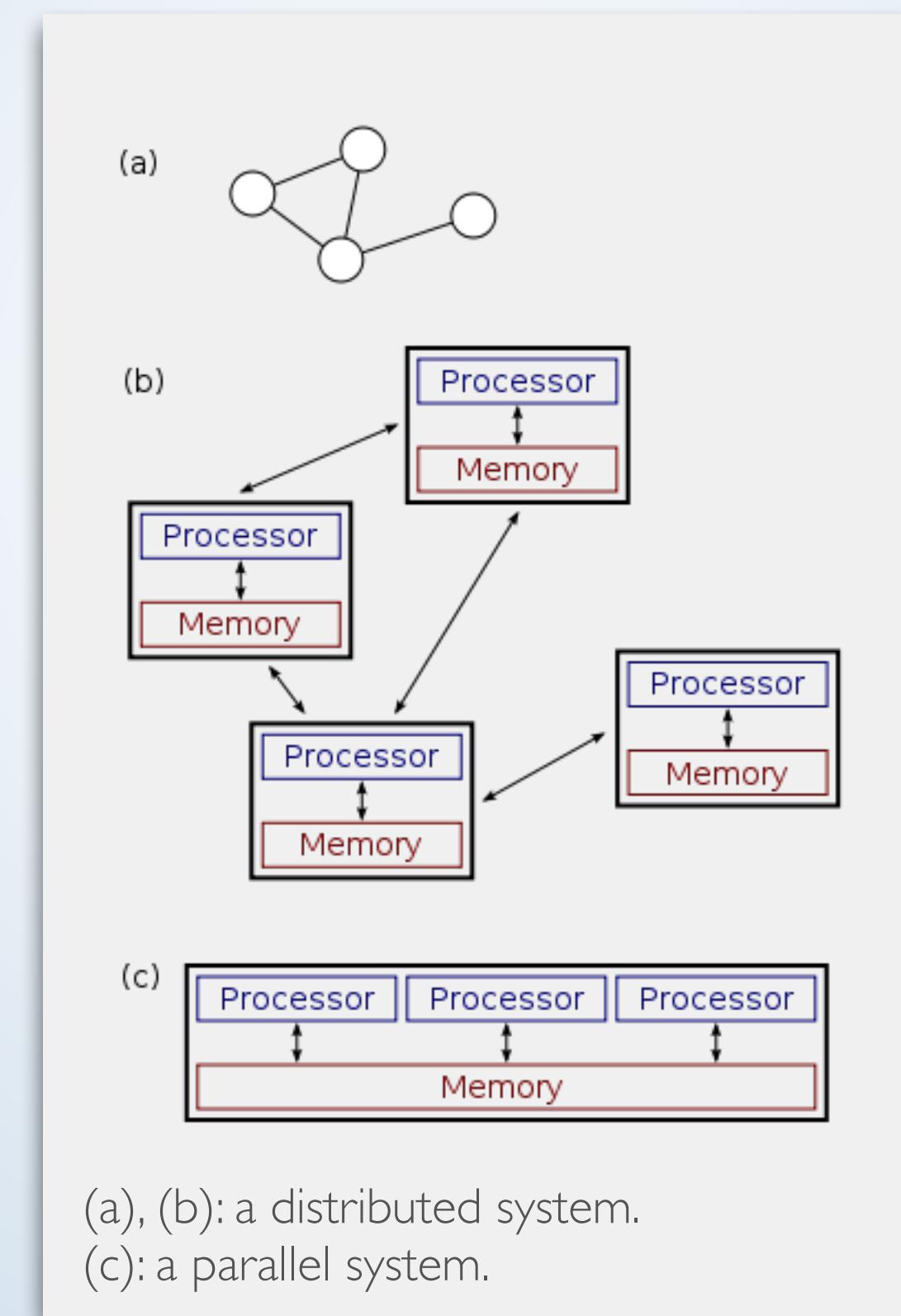
# Disconnect
dbDisconnect(db)
```



Importing Distributed Data

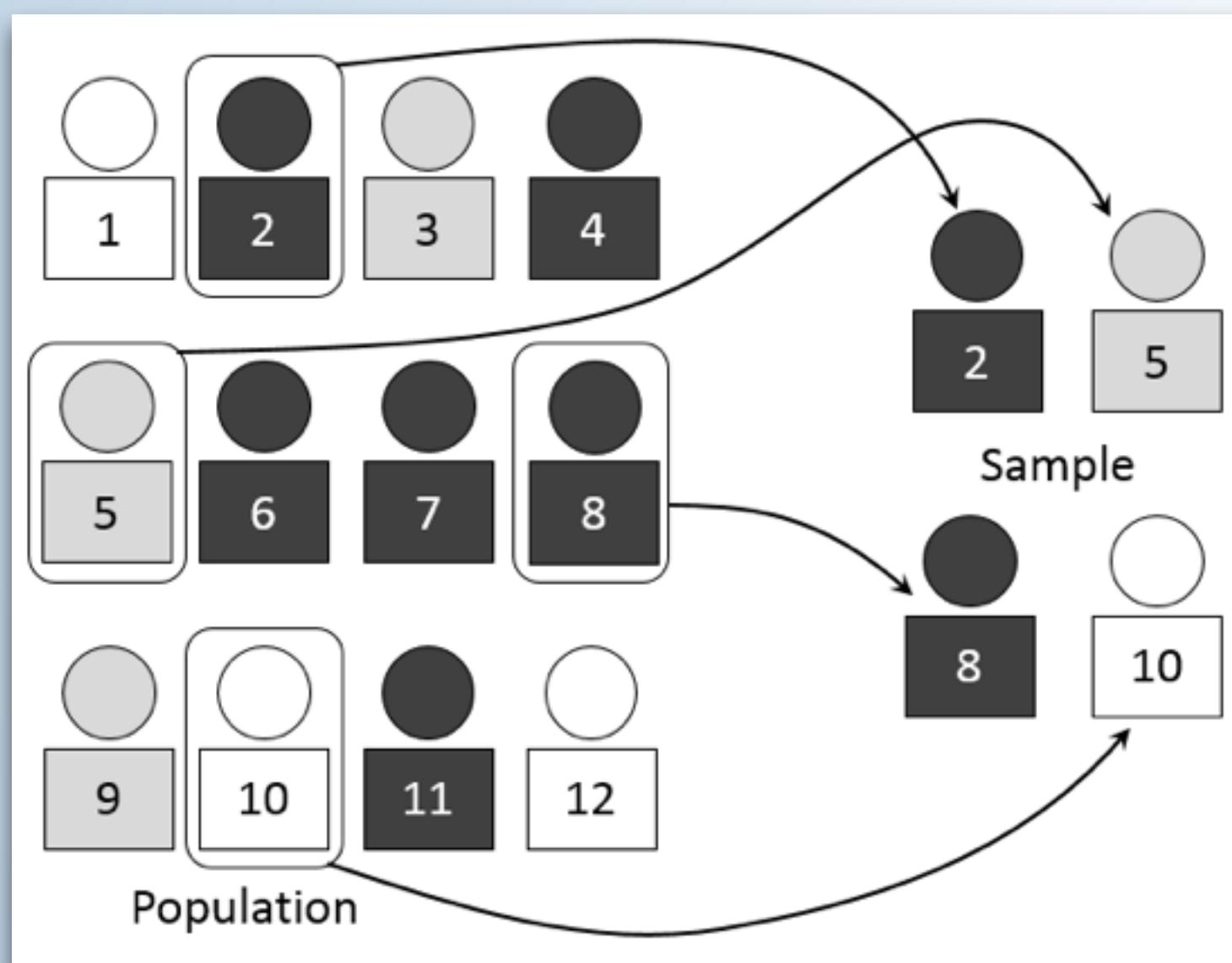
WHAT IS DISTRIBUTED DATA?

“non-relational with quick access to data over a large number of nodes” - Wikipedia



IMPORTING DISTRIBUTED DATA

Sampling



Distributed computing



